

REMARKS

This amendment responds to the Office Action mailed on November 19, 2004. Filed concurrently herewith is a *Request for a Two Month Extension of Time* which extends the shortened statutory period for response to expire on April 19, 2005. Accordingly, Applicants respectfully submit that this response is being timely filed.

Claims 1-12 were pending. In the Office Action mailed November 19, 2004, claims 1 and 9 have been amended to more clearly define protection to which Applicant is entitled, and new claim 13 is submitted for examination on its merits. Accordingly, claims 1-13 are pending in the present application, and Applicant believes these claims are in proper condition for allowance for the reasons set forth below.

§ 101 Rejections

Paragraphs 4-8 of the Office Action reject claims 1-12 under 35 U.S.C. § 101 as lacking patentable utility. Applicant respectfully traverses this rejection and request withdrawal of this rejection based on the following reasons.

It is asserted in the Office Action that the claimed “method for emulating a processor” in claims 1-4, the “process for compiling or translating a computer program instruction” in claim 5 and the “endian transformation system” in claims 9-12 are not directed toward statutory material because certain operations in these claims are within the capabilities of a human being, even though each of the claims themselves recite physical computer components for performing such operations (i.e., processors, bytes stored in memory, etc.). Initially, Applicants note that the Examiner’s interpretation of 35 U.S.C. §101 statutory subject matter is misguided when asserting that processes capable of being performed by a human being are nonstatutory. The Federal Circuit has held that “[t]he inclusion in a patent of a process that may be performed by a person, but that also is capable of being performed by a machine, is not fatal to patentability.” *Alco Standard Corp. v. Tennessee Valley Authority*, 808 F.2d 1490, 1496, 1 U.S.P.Q.2d 1337, 1337 (Fed.Cir. 1986). In the present situation, the asserted fact that applicants operations may be performed by a person, even though capable of being performed by a processor, does not render the subject matter nonstatutory.

Still further, Applicant respectfully submits that the claimed invention produces a useful, concrete and tangible result. The claims clearly recite that they produce the useful and tangible result of transforming the memory access addresses such that bytes stored in memory addressed by a processor of a first endian format are observed as if the memory was addressed by a processor of a second endian format. These address transformations produce the useful, concrete and tangible result of ensuring that bytes aggregate in the memory in a pattern which is a mirror image of the pattern which would have resulted if the processor had been the other endian format and no address transformation had been performed. Thus, Applicant submits that address transformation performed on the bytes stored in memory is a useful, concrete and tangible result, and it is respectfully requested the §101 rejections of claims 1-12 be withdrawn.

Provisional Double-Patenting Rejection

Paragraph 11 of the Office Action provisionally rejected claims 1-12 under 35 U.S.C. § 101 as claiming the same invention as that of claims 1-12 of co-pending U.S. Patent Application Serial Nos. 10/176,694 and 10/177,131. Applicant respectfully submits that this double patenting rejection is moot since preliminary amendments canceling Claims 1-12 were filed in each of these co-pending applications. Accordingly, Applicant respectfully requests withdrawal of the rejection.

Claim Rejections Under 35 U.S.C. §112, First Paragraph

Paragraph 14 of the Office Action rejects claims 1-4 and 9-12 under 35 U.S.C. § 112, first paragraph, as not being enabled for every possible conceivable means of emulating a processor including those that translate memory addresses. Applicant traverses this rejection and respectfully requests reconsideration based on the following remarks.

Initially, Applicant notes that the requirements for rejecting claims under §112 for lacking enablement commensurate with the scope of the claims has not been set forth in the Office Action. M.P.E.P. § 2164.08 recites that “[i]f a rejection is made based on the view that the enablement is not commensurate in scope with the claim, the examiner should identify the subject matter that is considered to be enabled. By asserting that the disclosure is “enabling for

at most those means of emulating a processor known to the inventor,” the subject matter considered to be enabled has not specifically been set forth in the Office Action.

Further, Applicant respectfully submits that the specification clearly teaches those skilled in the art how to make and use the full scope of the claimed invention without 'undue experimentation'." *In re Wright*, 999 F.2d 1557, 1561, 27 USPQ2d 1510, 1513 (Fed. Cir. 1993). For instance, the Examiner admits in paragraph 14 of the Office Action that the “process of emulating a processor which consists solely of translating memory addresses with regard to big or little endian arrangements” is enabled by Applicant’s written description. However, this is the very feature of translating memory addresses from one endian format to another endian format is recited in claims 3 and 11. Page 8, lines 6-7 of the present specification recites that “the invention can be used to transform between any two endian systems which are byte reversals of one another.” Clearly, claims 3 and 11 are enabled commensurate with their scope.

Claims 1 and 9 are directed to an endian transformation method and system in which memory access addresses are transformed such that the pattern of bytes stored in a memory addressed by a processor observing a first convention for ordering of the significance of bytes in a word (e.g., big endian) is a ‘mirror image’ of the pattern which would have resulted if the memory had been addressed by a processor observing a first convention for ordering of the significance of bytes in a word (e.g., little endian). This is clearly described throughout the present specification, including from page 4, penultimate line – page 5, line 2). Clearly, this ‘mirror image’ feature of claims 1 and 9 is enabled commensurate with their scope.

Claims 2 and 10 are directed to an endian transformation method and system in which memory access addresses are transformed such that the pattern of bytes stored in a memory addressed by a processor observing a first convention for ordering of the significance of bytes in a word (e.g., big endian) is in the ‘reverse order’ of the pattern which would have resulted if the memory had been addressed by a processor observing a first convention for ordering of the significance of bytes in a word (e.g., little endian). This is clearly described throughout the present specification, including the paragraph extending between pages 4-5. Clearly, this ‘reverse order’ feature of claims 2 and 10 is enabled commensurate with their scope.

Claims 4 and 12 are directed to an endian transformation method and system of emulating a processor observing a first convention for ordering of the significance of bytes within words on a second type of processor observing a second convention for ordering of the

significance of bytes, wherein each memory access address B of string length L is transformed to the address A-B-L+S, wherein A is the total number of bytes allocated to a program, and S is the start address of the program. This is clearly described throughout the present specification, including page 2, lines 21-26. Clearly, this 'A-B-L+S' feature of claims 4 and 12 is enabled commensurate with their scope.

From the foregoing it can be seen that every element appearing in the claims has been described and enabled in the present specification. Applicant notes that how a teaching is set forth, by specific example or broad terminology, is not important. *In re Marzocchi*, 439 F.2d 220, 223-24 169 USPQ 367, 370 (CCPA 1971). When analyzing the enabled scope of a claim, the teachings of the specification must not be ignored because claims are to be given their broadest reasonable interpretation that is consistent with the specification. "That claims are interpreted in light of the specification does not mean that everything in the specification must be read into the claims." *Raytheon Co. v. Roper Corp.*, 724 F.2d 951, 957, 220 USPQ 592, 597 (Fed. Cir. 1983), cert. denied, 469 U.S. 835 (1984). The scope of claims 1-4 and 9-12 is consistent with their description in the specification and Applicant is not required to solely limit his claims to a scope of translating memory addresses between big and little endian arrangements. Applicant respectfully requests that claims 1-4 and 9-12 are properly enabled and would not require 'undue experimentation' to teach those skilled in the art how to make and use the full scope of the claimed invention. Accordingly, Applicant requests withdrawal of the § 112, first paragraph, rejections of claims 1-4 and 9-12.

Claim Rejections Under 35 U.S.C. §112, Second Paragraph

Paragraphs 16-17 of the Office Action reject claims 1-4 under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to provide a definition of the term "emulate" in the specification. Applicant respectfully traverses this rejection and requests its immediate withdrawal based on the following remarks.

Applicant notes that the meaning of the term "emulating" appearing in the claims and specification of the present application is commensurate with the meaning understood to those skilled in the art. In fact, the dictionary definition provided by the Examiner from the Microsoft

Computer Dictionary fully coincides with the definition provided by the Applicant in the present specification:

Microsoft Computer Dictionary	Applicant's Specification
Emulate vb. For a hardware or software system to behave in the same manner as another hardware or software system.	Emulation systems enable software of one endian format to operate on hardware of an opposite endian format (page 1, lines 16-18).
Emulator n. Hardware or software designed to make one type of computer or component act as if it were another. By means of an emulator, <i>a computer can run software written for another machine.</i>	The present invention provides an efficient (emulating) method and system to <i>enable software of one endian format to run on hardware of a different endian format</i> (page 1, lines 22-24).

Clearly, it can be seen that both Applicant's specification and the dictionary definitions cited by the Examiner both define the term "emulate" means to enable software designed for one type of computer hardware to operate on hardware of a different format. Thus, Applicant's definition of 'emulating' and the definition understood by those skilled in the art coincide. Applicant respectfully submits that claims 1-4 are not indefinite and requests immediate withdrawal of the § 112 indefiniteness rejection.

Claim Rejections Under 35 U.S.C. §103

Paragraph 20 of the Office Action rejects claims 9-12 under U.S.C. § 102(b) as being anticipated by U.S. Patent No. 5,968,164 to Loen et al. (hereinafter "Loen"). Applicant respectfully traverses this rejection and submits that the claims at issue are patentable over the cited prior art for the following reasons.

It is noted that only claims 9-12 are rejected but the Examiner discusses claims 1-8 in paragraphs 21-28. Applicant will address the Examiner's comments in paragraphs 21-28 assuming it was the Examiner's intention to reject claims 1-8 as well.

Claims 1-12 are directed to an endian transformation method and system that enable software of one endian format to run on hardware of a different endian format. As described on page 1, lines 18-21 of the present specification, generally, prior systems of this type converted each word between endian representations on a word-by-word basis. When such conversions

were frequently required, a significant overhead into the time required to perform a given task was introduced. Loen appears to disclose such a prior art system. In order to reduce this overhead and provide considerable time savings, Applicant invented the endian transformation method system embodied in claims 1-12 where address transformations are performed to an entire address space in simple one-step arithmetic operations to ensure that bytes aggregate in the memory in a pattern that is a reversal or mirror image of the pattern which would have resulted if the processor had had an opposite endian format.

Loen Fails to Disclose Creating a Mirror Image of the Entire Address Space

Independent claims 1 and 9 have been amended above to recite that memory access addresses for the entire address space are transformed bytes aggregate in memory in a byte order which is a mirror image of the distribution pattern of bytes which would have resulted if the memory was addressed by a processor of a type having a different convention for ordering the significance of bytes within words.

Rather than transform the entire address space in this manner, Loen discloses a mixed-endian computer system that allow tasks having different format expectations (i.e., big versus little endian) to co-exist and execute, on a task by task basis. The main memory 109 of Loen's computer system 100 includes both big endian data and little endian data, where an endian bit 177 is stored in a table entry to identify whether the data is big endian or little endian, *see column 10, line 62 – column 11, line 3*. When the endian of the processor does not match that of the software task, Loen discloses a two-step conversion process for converting the data between the two endian formats on a word-by-word basis by changing the byte order of a word or fragment thereof, *see column 7, lines 1-10*. For instance, Loen discloses that its endian conversion method can be used with 16-bit, 32-bit, 64-bit, 128-bit or other word sizes, *see column 8, lines 31-35*.

By performing this conversion on a word-by-word basis, the byte locations within a word that are transformed when converting from big endian to little endian, or vice versa, will only be the mirror images of each other within a specific word but not for the entire address space. By way of example, consider the following three words in the address space below:

0	1	2	3	4	5	6	7
a	b	c	d	e	f	g	h
8	9	10	11	12	13	14	15
i	j	k	l	m	n	o	p
16	17	18	19	20	21	22	23
q	r	s	t	u	v	w	x

Big Endian Address Space

0	1	2	3	4	5	6	7
h	g	f	e	d	c	b	a
8	9	10	11	12	13	14	15
p	o	n	m	l	k	j	i
16	17	18	19	20	21	22	23
x	w	v	u	t	s	r	q

**Little Endian Address Space
After Loen Conversion**

As illustrated in this example and similar to the example shown in FIG. 4A of Loen, the pattern of bytes stored in memory for each specific word would be in a byte order that is a mirror image of the pattern of bytes stored in memory for the other endian format. However, the pattern of bytes stored in memory for the entire address space would not be mirror images of each other when using a word-by-word conversion process. In contrast, using the same big endian address space from the above example, Applicant's transformed little endian address space would appear as follows:

0	1	2	3	4	5	6	7
a	b	c	d	e	f	g	h
8	9	10	11	12	13	14	15
i	j	k	l	m	n	o	p
16	17	18	19	20	21	22	23
q	r	s	t	u	v	w	x

Big Endian Address Space

0	1	2	3	4	5	6	7
x	w	v	u	t	s	r	q
8	9	10	11	12	13	14	15
p	o	n	m	l	k	j	i
16	17	18	19	20	21	22	23
h	g	f	e	d	c	b	a

**Applicant's 'Mirror Image'
Little Endian Address Space**

From this, it can be seen that Applicant's entire address space is a mirror image of the address space for the opposite endian format, whereas Loen only appears to create mirror images of bytes in specific words that are transformed. Accordingly, Applicant respectfully submits that

Loen fails to teach or suggest transforming memory access addresses for an entire address space such that the aggregate pattern of bytes stored in memory is a mirror image of the distribution pattern of bytes which would have resulted if the memory was addressed by a processor of a type having a different convention for ordering the significance of bytes within words. Thus, each and every feature of amended claims 1 and 9 are not disclosed by Loen, and Applicant requests that the § 102 rejection of claims 1 and 9 be withdrawn.

Loen Fails to Disclose Maintaining the Offset Between Any Two Bytes Stored in Memory After the Transformation

Independent claims 2 and 10 recite that the “memory access addresses are transformed such that the offset between addresses of any two bytes stored in memory is unaltered by the transformation and the relative order of the addresses of any two bytes stored in the memory is reversed by the transformation.” As illustrated in the example on the prior page, the offset between any two bytes stored in memory is unaltered by Applicant’s transformation, where the relative order of the addresses stored in memory is merely reversed by the transformation. For instance, bytes “p” and “q” in the example remain in adjacent memory locations after the transformation, where their relative order is reversed in the transformation. In the big endian address space, bytes “p” and “q” are respectively located in locations 15 and 16 in the address space. After the little endian transformation, it can be seen their offset remains the same but their order is reversed where “p” and “q” are respectively located in locations 7 and 6 in the little endian address space.

Contrarily, since Loen appears to convert between endian formats on a word-by-word basis, bytes “p” and “q” would respectively be located in locations 8 and 23 in the little endian address space according to Loen’s conversion process. From this, it can be seen that Loen fails to maintain both. 1) the offset between addresses of any two bytes stored in memory in an unaltered state by the transformation, 2) the relative order of the addresses of any two bytes stored in the memory in a reversed order after the transformation. Thus, Applicant submits that each of the claim limitations of claims 2 and 10 are not taught or suggested by Loen, and the § 102 rejections of these claims are requested to be withdrawn.

Loen Fails to Disclose Storing Successive Strings of Bytes That Have Been Transformed

With respect to independent claims 3 and 11, it is recognized in paragraph 23 of the Office Action that Loen fails to disclose the feature where “memory access addresses are transformed such that strings of bytes in the first endian format which are stored successively by the processor operating in accordance with the second endian format aggregate in the same manner as the bytes would aggregate if the processor was of the first endian format and memory access addresses were not transformed.”

The Examiner asserts that this feature is inherent in Loen because “if the invention of Loen et al. did not aggregate byte strings as though they had not been translated, then code would need to be specially designed for the processor to accommodate this peculiarity of the endian transformation.” Applicant traverses this reasoning and submits that Loen teaches quite the opposite effect. The disclosed system in Loen is designed to precisely handle such a peculiarity, because it converts between endian formats on a task-by-task basis so that it can dynamically change its endian mode (*see Abstract*). This dynamic conversion dictates that Loen performs conversion on a word-by-word basis, and, as shown above in the preceding examples, it is not inherent in Loen that it will store strings of bytes successively to aggregate in the same manner as the bytes would aggregate if the processor was of a different endian format and the memory access addresses were not transformed. In fact, in the example described above, it is shown above that strings of bytes are separated from each other after the endian conversion in Loen.

Thus, Applicant respectfully submits that the claimed features of independent claims 3 and 11 are not taught or suggested by Loen, and it is respectfully submitted that these claims are in proper condition for allowance. Reconsideration of the outstanding rejection is respectfully requested.

Loen Fails to Disclose Transforming Each Memory Access Address B of String Length L to the Address A-B-L+S

Independent claims 4 and 12 recites that each memory access address B of string length L is transformed to the address $A-B-L+S$, where A is the total number of bytes allocated to a program and S is the start address of the program. Applicant developed this address transformation operation to allow one extra arithmetic operation to be used with every load/store

instruction to simply transform memory access address between two different endian formats.

Contrarily, Loen discloses a two step process to convert between two different endian formats. As described in column 7, lines 5-10 and lines 35-40 of Loen:

The first step is a reflection which must be performed on the bytes comprising the data double word or fraction thereof (see FIG. 3a). The second step is a modification of the memory address offset of the bytes comprising the data double word to accommodate the new location of the bytes after the reflection that was performed in the first step (see FIG. 3b). . . . Referring to FIG. 3b, a preferred system performs an Exclusive-or (XOR) 7 operation during a 8-bit byte memory reference.

Thus, Loen requires that both a reflection of the bytes be performed and then an XOR operation be performed on the reflected bytes in order to convert between the two endian formats. Loen fails to teach or suggest transforming a memory access address B of string length L to a new address using the simple arithmetic operation $A-B-L+S$, where A is the total number of bytes allocated to a program and S is the start address of the program. The Examiner has selected these appropriate values from Loen's teachings and plugged them into Applicant's formula to illustrate that the final converted address would be the same. However, the Examiner is merely proving that Applicant's invention in fact works, because the endian transformation of a single memory access address should ultimately end up with the same transformed memory access address even if different endian transformation techniques were utilized to get there. The key difference being that Applicant's novel arithmetic operation $A-B-L+S$ allows the transformation to be completed in a single, simple step, as opposed to the two more complex reflection and XOR steps as performed by Loen.

Thus, Applicant submits that each of the claim limitations of claims 4 and 12 are not taught or suggested by Loen, and the § 102 rejections of these claims are requested to be withdrawn. Further, new claim 13 is submitted for examination on its merits to further exemplify this aspect of Applicant's endian transformation method.

With respect to independent claim 5, a process for compiling or translating a computer program code instruction is recited in which a referenced memory address with respect to a fixed block size of memory is transformed so as to change the referenced address value by an amount

that is fixed for a given number of bytes being accessed in a word. As described above, referenced memory address can simply be modified using a single arithmetic operation ($A-B-L+S$), where the fixed block size of memory (A), the given number of bytes being accessed in a word (L) and the start address (S) would be known and fixed for a given referenced memory address. Thus, the amount $A-L+S$ would be the fixed amount that the referenced address value would be changed. This feature is not taught or suggested by Loen, since Loen discloses a two-step conversion process including a reflection step and a separate XOR step. Thus, Applicant submits that each of the claim limitations of claims 5-8 are not taught or suggested by Loen, and the § 102 rejections of these claims are requested to be withdrawn


Conclusion

In view of the foregoing remarks and amendments, Applicants respectfully submit that the subject application is in condition for allowance. Applicants, therefore, respectfully request reconsideration and early notice of allowance.

Respectfully submitted,

PAUL, HASTINGS, JANOFSKY & WALKER LLP

Dated: 5/19/05

By: 
Bradley D. Blanche
Reg. No. 38,387

PAUL, HASTINGS, JANOFSKY & WALKER LLP

P.O. Box 919092

San Diego, CA 92191-9092

Phone: (858) 720-2500

Fax: (858) 720-2555